

Description

SYSTEM AND METHOD FOR REMOTE CONTROL OF SOFTWARE AND AN ATTACHED DEVICE

5

Cross-Reference To Related Application

This application claims priority from U.S. provisional application no. 60/268,592, filed February 13, 2001.

10

Field of the Invention

This invention relates to customer support within the computer and software industry and collaboration between geographically remote computer users. This invention particularly relates to the remote control of a software application and, where applicable, a hardware or software device attached to the host computer.

20

Background of the Invention

The growth of the Internet allows computers all over the world to be connected to one another via various networks. These connections allow computer users to communicate with one another for both business and personal reasons. It is also possible for one remotely-located user to view and control a host-user's computer. (In this application, the term "host" refers to the computer containing the software and, where applicable, the device whose control is shared with another "remote" computer.) Remote access and control of computers is useful for transferring files and system administration. It can also provide an opportunity for geographically separated coworkers to collaborate on projects. Computer-to-computer customer service and technical support are also enabled by remote access and control technology.

The remote-control collaboration/customer support technology would be particularly useful in the

SCANNED, #24

development of microcontrollers, integrated chips designed to control specific systems. Microcontrollers run application-specific software which engineers must debug during the hardware/software integration phase of development. Emulators, such as those available from NOHAU Corporation, have been developed to simplify the debugging process by providing engineers an interface to the microcontroller. The microcontroller is attached to the engineer's computer as a hardware device; the engineer can then run the application-specific software intended for the microcontroller and debug the software and validate the integrated (hardware and software) microcontroller design. Given the technical details involved in running these emulators, a high level of technical support is desirable. It would be very useful for a technical support specialist at a remote location to be able to view the customer's monitor and control the customer's software application and the attached device. Similarly, a system where collaborating engineers could remotely view and control an emulator running on a host computer would also be very useful. The current invention seeks to improve upon the prior art relating to remote control of software applications.

The Telnet protocol allows remote access of a host computer. Remote users may access the host computer via telnet, a text-based protocol, as long as they can successfully log in with a user ID and password. Upon successfully logging in, the remote user can access specific applications or data located on the host computer.

Remote users may request files from other computers via other protocols such as the File Transfer Protocol (FTP) and the Hypertext Transfer Protocol (HTTP). However, these protocols grant access to files only, not the entire computer.

U.S. Patent No. 5,909,545 "Method and System for On Demand Downloading of Module to Enable Remote Control of an Application Program over a Network"

discloses a method of demonstrating and remotely
controlling software over a network via a remote display
module transported from a host computer to a guest
computer. The remote display module is executed at the
5 guest's computer to establish communication between the
interface to computer resources at the guest's and host's
computer. The application and an application
interception module are launched at the host computer to
establish communication between the application
10 interception module and the remote display module. This
allows I/O messages to be communicated between the
application and the user interface at the guest computer.
The remote display module converts remote control
messages and responses to a format understood by the host
15 application and the guest computer, allowing output to be
provided to the guest and the guest's input to be
provided to the host application. The preferred
embodiment is to present the modules and application
program through HTTP servers to a guest system which uses
20 a browser having a JAVA interpreter to execute the remote
display module and convert the remote control protocol
messages.

U.S. Patent Nos. 5,801,689 "Hypertext Based
Remote Graphic User Interface Control System" and
25 5,949,412 "Computer Remote Control System" disclose a
method of controlling a remote computer via the World
Wide Web. There are two main components of these
inventions - a GUI-screen-to-hypertext converter and a
hypertext-to-GUI response means. These two components
30 allow the host computer's screen to appear in a web page
where it can be controlled by the remote user.

There are some commercial remote control
software products. These products include PC Anywhere,
Laplink, Carbon Copy, and Double Vision. These products
35 control the entire computer, not a single application.
In addition, these are "external" applications which will
operate slower than a solution that is native to the

software application being used by the host and remote users. None of these products is able to remotely control a hardware device attached to a host computer.

There is no discussion in the prior art of how
5 to remotely control a single application rather than the entire computer. Similarly, there is no discussion of how to remotely control a hardware device attached to a host user's computer. There is also no discussion of how to enable remote control of a host computer by a remote
10 computer without compromising the host organization's internet security. The prior art also does not discuss how to accommodate "double control" where both the host and remote users can control the device. Finally, there is no discussion of remote control of a computer or
15 software application without the use of HTML, an Internet browser, or operating system-related messages.

It is an object of the invention to enable remote control of a software application via communication that is native to the application.

20 It is an object of the invention to enable remote control of a software application by a remote user in another organization that does not compromise the host organization's Internet security.

It is another object of the invention to enable
25 remote control of a single software application that does not require control of the entire host computer.

It is another object of the invention to enable remote control of a software application without relying on HTML, an Internet browser, and operating system-
30 related commands or messages.

It is an object of the invention to enable a remote computer user to view and control software running on a host computer.

It is another object of this invention to
35 enable a remote computer user to view and control software and a hardware device controlled by the software which is running on a host computer.

It is another object of this invention to provide an improved approach to customer support for software applications.

5 It is another object of this invention to provide an improved method for collaborations among users of a particular software application.

Summary of the Invention

10 This invention provides a system and method that allows both a remote computer user and a host computer user to view and control a software application running on a host computer. In instances where the host's software application controls a device connected to the host computer, the invention enables the remote
15 user to view and control the operations of that device.

The host and remote computers share information and control of the software and attached device via the software application's graphical user interface. The host and remote applications are linked via a TCP/IP
20 connection between the software applications. Using this approach, only the applications, and not the computers as a whole, are linked.

Before a connection can be established, one of the software applications should be configured as
25 "client-side" and one should be configured as "server-side." (The terms "client" and "server" have their usual definitions in this application, i.e., a client requests the services of and accepts the responses of a server.) If the host, but not the remote user, is located behind a
30 firewall, the host's software should be configured as "client-side" while the remote computer's software should be configured as "server-side" since the remote user would ordinarily not be able to initiate the connection due to the firewall.

35 The invention allows multiple users whose software application is configured to be client-side to be connected simultaneously to a computer configured as a

server. The device the software application controls can be attached either to the server or to any of the clients.

Once a connection is established, any action performed by either the host or remote application will control the other side of the connection as well as any hardware device that may be connected to one of the computers. In addition, when a hardware device is connected to one of the computers, all device events can be monitored via the graphical user interface of the host and remote applications.

Brief Description of the Drawings

Fig. 1 is a block diagram showing the configuration of a customer support system in accordance with the invention.

Fig. 2 is a diagram showing the components of the host and remote software applications in accordance with the invention of Fig. 1.

Fig. 3 is a flowchart showing how commands issued by a host user may be viewed by a remote user in accordance with the invention.

Fig. 4 is a flowchart showing how commands issued by a remote user may be viewed by the host user in accordance with the invention.

Fig. 5 is a block diagram showing the configuration for a customer linked to two support computers in accordance with the invention.

Fig. 6 is a block diagram showing a possible configuration for geographically remote collaborators who are linked via a software application in accordance with the invention.

Detailed Description of the Invention

With reference to Fig. 1, host computer 10, running a software application 20 which controls the attached device 12, is located behind a firewall 14. Possible devices 12 include a hardware device such as an

in-circuit emulator, a microprocessor, a data storage device, a test instrument, or an internet-enabled device. A software device such as a simulator may also be attached. The host 10 user decides to seek help from a customer support specialist who is at a remote computer 18. The term "host" will refer to the software application being viewed and controlled by another software application. The term "remote" will refer to the application viewing and controlling the host application. Remote computer 18 has a copy of the same software application 20 that is running on the host computer 10. In order to get help from the customer support specialist at remote computer 18, the user at the host computer 10 initiates a connection 22 via the Internet 16 between the host computer 10 software application 20 and the remote computer 18 application 20.

In the customer support context, a connection between a customer and a support specialist must be established. If a customer is behind a firewall, the customer's application must be configured as client-side software; the support specialist's application must therefore be configured as server-side software. Although Fig. 1 shows the device 12 attached to the computer whose application is configured as a client, the device 12 could be attached to a computer whose application is configured as a server. Only one device 12 may be attached to the application-linked computers. The only time an application must be configured as client-side software is if the computer on which it is running is behind a firewall or a similar security device. An application installed on a computer which is behind a firewall or similar security device cannot be designated as a server unless the IP address requesting a connection has permission to get behind the firewall (i.e., the source address is acceptable).

The customer will establish a connection by choosing the "Support Over IP" option from the "Help" menu, entering the support computer's IP address, and

requesting that an optional password-protected connection be established. A TCP/IP connection is established between the two applications. Any action by either of the applications will control the application on the other side of the connection as well as the hardware device attached to the customer's computer. Either the customer or the support technician may terminate the connection at any time.

In Fig. 2, a host (customer) computer 10 and a remote (support) computer 18 are running the same instances of a software application 20, 34 which have the same components but are configured slightly differently. Arrows indicate the paths messages may take within and between the components of the applications 20, 34. The two software applications 20, 34 are connected via a TCP/IP connection 22. A hardware or software device 12 may be attached to the host computer 10. In the host application 20, a core component 24 contains most of the code for the software's operations and, where a hardware or software device 12 is attached to the computer 10, controls the hardware or software device 12. A transmit thread (T1) 56 in the core component 24 sends commands back to the graphical user interface (GUI) 30 which allows the viewer to input commands or view the output of the application 20. A communication module (CM) 26 is responsible for passing messages between the core component 24 and the graphical user interface 30. A queue (Q) 48 in the communication module 26 holds messages to be passed to the graphical user interface 30, while a thread (T2) 52 takes messages from the queue 48 and passes them to the graphical user interface 30. Messages can include commands, requests for information, replies to requests, and notifications of events that have occurred to the hardware or software device 12, if any is attached. The application 20 also contains a client/server module (CSM) 28. The client/server module

has a socket (S) 32 which allows for bi-directional communication between applications. A read thread (T3) 54 takes messages out of the socket 32.

The remote computer 18 application 34 contains the same components with similar functions. A core component 36 contains much of the code for the software application 34. The core component 36 also contains a transmit thread (T1) 62 for sending messages from the core component 36; however, when the application 34 is connected to the host application 20, neither the core component 36 or the transmit thread 62 are employed since the host application's 20 core component 24 is active. A graphical user interface (GUI) 44 is included along with a communication module (CM) 42 (containing a queue (Q) 50 and a thread (T2) 58 for taking messages from the queue 50 to the graphical user interface 44) and a client/server module (CSM) 38 which contains a socket (S) 40 allowing bi-directional communication between applications. A read thread (T3) 60 takes messages out of the socket 40 and passes them to the queue 50 in the communication module 42.

When the applications 20, 34 are connected, both the host and remote graphical user interfaces 30, 44 can send two types of commands to the host core component 24. These are: 1) non-wait commands, which are transmitted to provide some instruction to either the core component 24 or the hardware device 12 that may or may not return a result at a future time; and 2) wait commands, which are transmitted and require an answer before the applications 20, 34 are allowed to continue their execution. A time-out can be set for a wait command. Where wait commands are issued by the graphical user interfaces 20, 34, the core component 24 must respond before other operations can occur. A blocking thread (T4) 64 in the host communication module and a blocking thread (T4) 66 in the remote communication module will monitor whether the core component 24 has responded to a wait command.

With reference to Fig. 3, commands entered by the host user may be viewed by the remote user after a connection between the two applications is established. Reference is also made to Fig. 2. The host user enters a
5 command via the host graphical user interface 30 (step 74). This command is sent from the host graphical user interface 30 to the host core component 24 via the host communications module 26 (step 76); at the same time, the
10 command is passed from the host graphical user interface 30 to the remote application 34 via the bi-directional sockets 32, 40 (step 78). The remote read thread 60 takes the message from the socket 40 and passes it to the queue 50 in the remote communication module 42 (step 80). A thread 58 picks the message out of the queue 50 and
15 sends it to the remote graphical user interface 44 (step 82). The remote user is now able to view the action taken by the host user.

When the host core component 24 responds to the command sent by the host graphical user interface 30
20 (step 84), a transmit thread 56 sends the message to a queue 48 in the host communication module 26 (step 86). Another thread 52 then picks the message out of the queue 48 and sends it to the host graphical user interface 30 (step 88), thus notifying the host user that the command
25 has been carried out. At the same time steps 86 and 88 are carried out, the remote user is also notified. The host transmit thread 56 sends a response to the remote application 34 via the bi-directional sockets 40, 32 (step 90). The remote read thread 60 takes this message
30 from the socket 40 and passes it to the queue 50 in the remote communications module 42 (step 92). A thread 58 picks this message out of the queue 50 and sends it to the remote graphical user interface 44 (step 94). This informs the remote user of the response to the command.

35 With reference to Fig. 4, commands entered by the remote user are relayed to the host application and may be viewed by the host user after a connection between the two applications is established. Reference is also

made to Fig. 2. A remote user enters a command via the remote application's 34 graphical user interface 44 (step 96). The command is passed to the host application 20 via the remote communication module 42 and bi-directional sockets 40, 32 (step 98). The host read thread 54 then takes the command from the socket 32 (step 100). The command is passed to the queue 48 in the host communication module 26 (step 102) where a thread 52 picks the command out of the queue 48 and sends it to the host graphical user interface 30 (step 104). This allows the host user to view the remote user's activity. While steps 102 and 104 are carried out, the command is also passed from the socket 32 to the host core component 24 via the host communication module 26 (step 106).

The host core component 24 responds to the command (step 108). Both the remote and host users will then receive a message from the host core component 24. The host transmit thread 56 sends a message to the queue 48 in the host communication module 26 (step 110). The message is picked out of the queue 48 by a thread 52 and sent to the host graphical user interface 30 (step 112). While steps 110 and 112 are carried out, the remote application 34 receives a message sent by the host transmit thread 56 via the bi-directional sockets 32, 40 (step 114). The remote read thread 60 takes the message from the socket 40 and passes it to the queue 50 in the remote communication module 42 (step 116). The thread 58 picks the message out of the queue 50 and sends it to the remote graphical user interface 44 (step 118).

As shown in Fig. 5, the customer support embodiment of the invention allows a customer's computer's 10 software application 20 to be linked via the Internet 16 with multiple support computers' 18, 68 software applications 34, 120. Before establishing the connection, the new customer support application 120 must be configured to reflect that it has no device 12 attached. (In the current embodiment, this can be done on the command line to start the application.) This new

application 120 would also have to be configured to be client-side software since connections are established with the application configured as a server. The new customer support application 120 user would then specify
5 the TCP/IP address of the original support computer and request that a connection be established. Once the connection is established, the customer support specialist using the second support computer 68 will be able to interact with the software 20 and any attached
10 device 12 with equal priority to the other two applications 20, 34. Messages are passed between the linked applications 20, 34, and 120 as shown Figs. 2, 3, and 4. The number of customer support specialists who may be linked is limited only by the application's
15 programming. Although the device 12 is described here and shown in Fig. 5 as attached to the customer's computer 10, the device 12 can be attached to either a server or one of the multiple clients (provided only one client indicates that it attached to a hardware device
20 12).

In Fig. 6, an alternative embodiment is shown where multiple collaborators at geographically remote locations can be linked via the software applications 122, 124, 128 running on their computers 70, 72, 126.
25 One of the collaborators' applications 122, 124, 128 should be configured as server-side software while the others should be configured as clients. Connections are established as discussed above and messages are passed as shown in Figs. 2, 3 and 4.

The preceding discussion has focused on the remote control of a software application and an attached hardware or software device. However, the concepts discussed above are applicable to remote control of a software application even where no hardware or software
30 device is attached to the host computer.

In another embodiment, the invention can be implemented with a predominantly message-driven architecture instead of the read and transmit threads

described above. While at least one thread still must be used to monitor whether the core component has responded to a wait command, messages may be passed among different components and modules without the use of threads when a message-driven architecture is employed.

5

090223-070901